An Efficient Semantic Web Service Discovery Using Hybrid Matching

Pensri Pukkasenung
Department of Computer Science
Faculty of Science and Technology, Rajabhat Rajanagarindra University
pensriash@hotmail.com

Peraphon Sophatsathit, and Chidchanok Lursinsap
Advanced Virtual and Intelligent Computing (AVIC)
Department of Mathematics, Faculty of Science, Chulalongkorn University
Bangkok 10330, Thailand

Abstract

This paper focuses on increasing the efficiency of a discovery Semantic Web Service (SWS) in the UDDI repository. We calculate similarity matching of web services in terms of capabilities in the service profile. The properties involved in the computation include input, output, precondition and effect (IOPE). We introduce the semantic hybrid matching that matches between web service advertisements and web service requests based on semantic distance and cosine law. Matching is repeatedly adjusted until the similarity measure satisfies with the right web service.

Keywords: web service, semantic web, service discovery, hybrid matching.

1. Introduction

The proliferation of WWW has brought about Web Intelligent or Semantic Web Service discovery which poses a challenging issue in service-oriented computing. Current web service architecture allows service providers to create WSDL descriptions and publish them to UDDI Registry where service requestors can look for their required services in UDDI repositories. However, there are two crucial limitations in UDDI registry for web service discovery mechanism. First, discovery web service classification mechanism would yield coarse results with high precision and recall errors. Second, UDDI guarantees syntactic interoperability, but fails to provide a semantic description of its content. In overcoming these limitations using semantic web technology, the capability of UDDI service registries is enhanced by applying matching algorithms between advertisements and requests described in OWL-S to recognize various degree of matching services.

In this paper, we present a hybrid matching that utilizes semantic distance and cosine law. The algorithm calculates similarity matching between web services from the requestor and web services published in UDDI repository. We define five matching categories, namely, exact, plugin, subsume, partial, and fail. We found that our definition adequately classified service matching. The experiment compared our proposed hybrid matching with actual web service search in a hotel reservation case study.

The organization of this paper is as follows. Some related works are briefly described in Section 2. Section 3 gives background information on semantic web services. Section 4 describes the proposed hybrid matching filters. Experimental results and some final thoughts are given Section 5 and 6, respectively.

2. Related work

There are various approaches to locate web services on the Web. Many researches focused on semantic discovery problem using hybrid matching method of web services based on OWL-S and capability descriptions, i.e., inputs, outputs, preconditions and effects (IOPE) of a service [10]. A similar approach [11] introduces an automatic discovery algorithm of semantic web service whose properties of web services include input, output, and matching algorithm using the cosine law to determine the similarity of web services under investigation.

Yet another approach used semantic distance information based on OWL-S [5]. The main contribution of this approach is the use of information on ontological representation of parent-child relation with path-length values which are defined as semantic distance information. Nevertheless, it utilized only input and output matching having 4 matching categories, namely, exact, plugin, subsume, and fail [5]. Fabien, et al, [1] presented an automatic method to calculate the similarity measure between two schema elements for discovering semantic web services. Approxivect [1] was based on the approximation of terminological method and on the cosine measure between context vectors.

We analyze the above work and present a new hybrid matching for efficient semantic web service discovery

3. Semantic Web Service

This section will describe some background of related technologies on semantic web service discovery.

3.1 OWL-S

OWL-S is an OWL-based web service ontology which supplies web service providers with a core set of markup language, constructed for describing the properties and capabilities of their web services in an unambiguous, computer-interpretable form [12]. The objective of OWL-S is to semantically describe web services for discovery and invocation. The structure of OWL-S consists of three ontology artifacts, namely, Service Profile, Process Model, and Grounding. The properties of this service class are:

Presents: the Service Profile provides detailed description of the service and its provider in a human readable way, i.e., input, output, precondition, and effect.

Describedby: the Service Model describes what the service does, how it works, and what functionality it provides as a process.

Supportedby: the Service Grounding provides information about service access specifications, such as communication and transportation protocols.

3.2 Semantic Web Service Discovery

Web Services technology realizes the basic service oriented architecture (SOA) which comprises three kinds of participants: service provider, service consumer, and service discovery agent or matchmaker. Enhancing service discovery with semantics can be achieved by introducing a semantic matchmaker. This semantic matchmaker will be aware of the semantic knowledge that is added to enrich service metadata. Such semantics can be represented by a semantic markup language such as OWL for used in discovery purpose. Descriptions of a service can be seen as comprising standard model descriptions and additional semantic descriptions. For web services technology, the association between a semantic description and the information entry of a web service in the standard UDDI can be maintained via model specification to which the service refers [9]. A semantic consumer can query a web service by interacting with the semantic matchmaker and specifying a semantic based request.

4. Hybrid matching algorithm

The underlying principle of hybrid matching is to calculate similarity matching between two web services of certain matching degrees. This method increases flexibility and availability of demands for web services required by the discovery mechanism to locate the desired web service. The procedures are carried out as described below.

4.1 Matching Definitions

An extended semantic matching degree consists of five filters, namely, exact, plugin, subsume, partial and fail matching. Such an establishment furnishes a finer-grained of service matching measure. Two kinds of services are considered, WsRequest and WsAdvertisement. WsRequest is a web service issued by the user and WsAdvertisement is the web service published in UDDI. The matching definitions are defined below.

- **Exact Match** if WsAdvertisement A and WsRequest R are conceptually equivalent.
- **Plugin Match** if WsRequest R is a sub-concept of WsAdvertisement A, notationally A > R.
- **Subsume Match** if WsRequest R is a super-concept of WsAdvertisement A or R>A.
- **Partial Match** if WsRequest R overlaps with WsAdvertisement A, A is not equal to R.
- **Fail Match** if WsRequest R unrelates to WsAdvertisement.

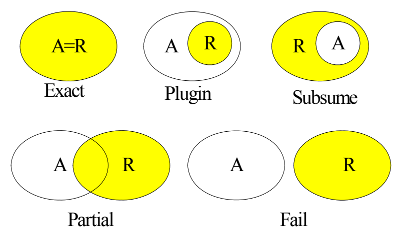

Figure 1. Illustration of matching definitions

4.2 Semantic Distance Weight

A semantic distance is the path length between the information on the ontology tree representation. The algorithm builds on subsumption relation of input, output, precondition, and effect (IOPE) represented by a tree construct. The vertex represents the concept; an edge represents the relation between two concepts with a weight to designate the parent-child association. A presumption of this association is a uniform distribution of immediate parent-child hierarchy. We calculate the weight of semantic distance, aka path_weight, between parent node (super -concept) and child node (sub-concept) as:

path_weight = path-length(parent) * path-length(child)

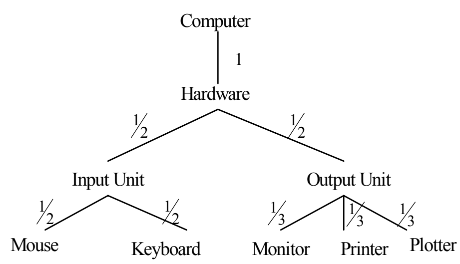

Figure 2. Ontology of typical hardware components in a computer

Figure 2 depicts an ontology of hardware components of a computer. The path_weight from Hardware to Monitor is 1/2*1/3 = 1/6

We applied this concept on input, output, precondition, and effect as the first step to calculate similarity of semantic web service between web service request and advertisement. Table 1 shows the path weight between the selected services

Table 1. Path_weight between 2 web services

Ws	Input	Output
WsRequest	Computer	Hardware
WsAdvertisement	Book	Printer
path_weight	1	1/2*1/3

4.3 Similarity Matching Algorithm

Define the set of input, output, precondition, and effect as follows:

$$WS = \{ I \;\; O \;\; P \;\; E \} \qquad (1)$$

where I denotes input of web services; O denotes output, P denotes precondition, and E denotes effect.

Data from each web service are arranged in a vector space to accommodate more than one parameter for some web services. To determine the similarity between any pair of vectors, we use cosine law to calculate the angle between them [11]. Denote web service advertisement in UDDI as WsA in Equation (2), and user request as WsR in Equation (3).

$$Ws_A = \{ i_A, o_A, p_A, e_A \} \qquad (2)$$

$$Ws_R = \{ i_R, o_R, p_R, e_R \} \qquad (3)$$

Cosine of the angle between WsA and WsR is given below.

$$\cos\theta = \frac{i_A i_R + o_A o_R + p_A p_R + e_A e_R}{\sqrt{i_A^2 + o_A^2 + p_A^2 + e_A^2} * \sqrt{i_R^2 + o_R^2 + p_R^2 + e_R^2}} \qquad (4)$$

If cosθ is equal to 1, WsA matches exactly with WsR. The smaller the value of cosθ, the farther WsA deviates from WsR. When cosθ is equal to 0, WsA will fail to match with WsR. We will set up the criteria for matching degree in the next section.

4.4 Scale of Matching Degree

Define the scale of matching degree from the result of cosine value obtained from Equation (4) as:

$$MD(WsR,WsA) = \begin{cases} \text{Exact} & \text{if } \cos\theta = 1.0 \\ \text{Plugin} & \text{if } \cos\theta \geq 0.8 \\ \text{Subsume} & \text{if } \cos\theta \geq 0.5 \\ \text{Partial} & \text{if } \cos\theta \geq 0.3 \\ \text{Fail} & \text{if } \cos\theta \leq 0.1 \end{cases}$$

This definition applies to the matching process as described by the state transition depicted in Figure 3.

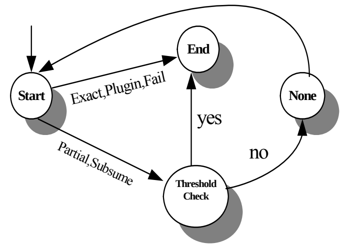

Figure 3. Matching transitions of web service

The transition states of web service matching consist of four states, namely, start, end, threshold check, and none. The start state begins when the cosθ of WsA and WsR are computed. If the matching degree falls within exact, plugin, or fail, transition moves to end state. Otherwise, check the threshold value. If it falls within the predefined threshold, the process ends in one of the above states. If this is not the case, it resorts to another WsA and repeats the process.

5. Experiment

Some preliminary experiments were conducted. First, we established the test data for web service Request (WsRequest) and web service Advertisement (WsAdvertisement). Next, determined the path_weight between WsRequest and WsAdvertisement for each IOPE component based on semantic distance. Computed web service similarity by cosine law and checked the similarity degree to establish the threshold values. Finally, we applied one WsRequest and three WsAdvertisements that were published in UDDI through a hotel reservation scenario. Assume that the user requested information about hotel booking using the following parameters:

WsRequest_Hotel = { City, Spa, Three Star, Creditcard, Confirmation, Visa, Booking }

WsAdvertisement_Hotel_1 = { Area, Hotel Facility, No.Star, Creditcard, Confirmation, Payment, Available }

WsAdvertisement_Hotel_2 = { Bangkok, Health club, -, - , Accept, Aeon, Confirm }

WsAsvertisement_Hotel_3 = { Area, SportClub, No.Star, Visa, Booking, Payment, - }

The service ontology artifacts were shown in Figure 4.

Table 2 shows the results of the experiment. The best matching having plugin match and 0.869 similarity value is WsAdvertisement_Hotel_1 matches with WsRequest_Hotel. The worst matching is fail match because all parameters of WsRequest_Hotel were irrelevant with WsAdvertisement_Hotel_2, and WsAdvertisement_Hotel_3 parameters.

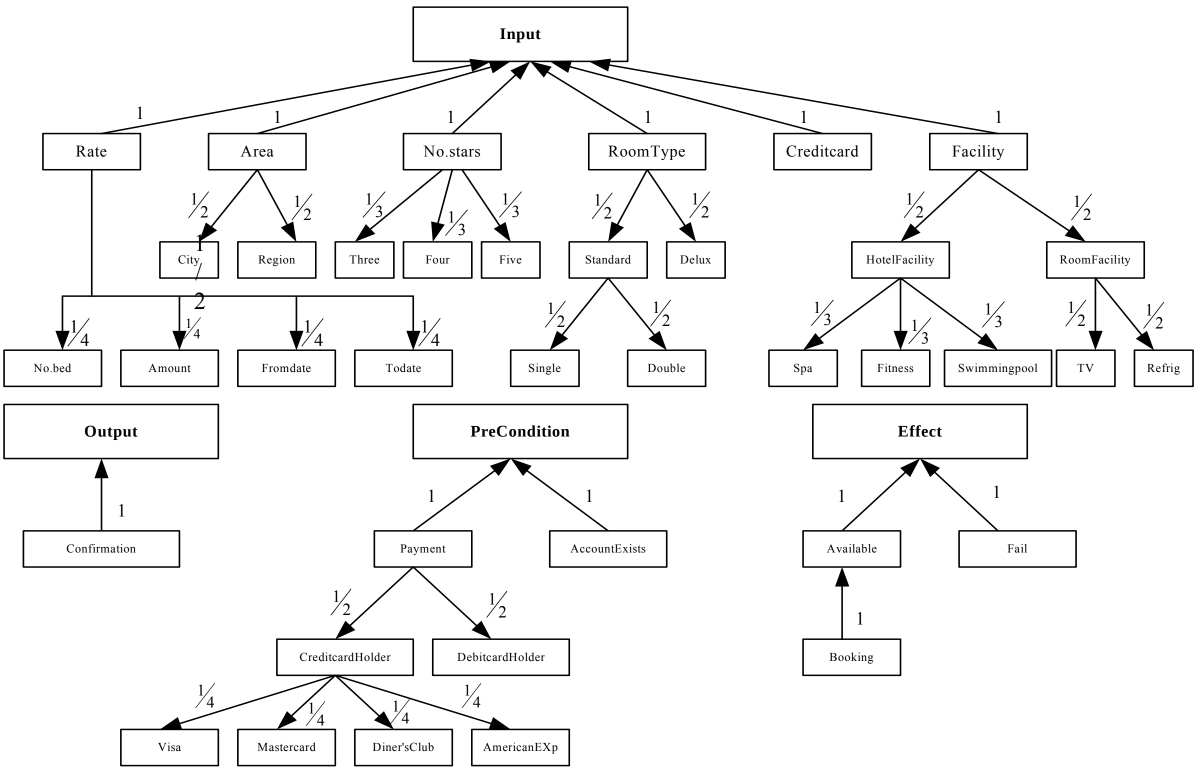

Figure 4. Ontology of The Hotel Reservation

Table 2. Path_Weight Between 2 Web Services

Ws	Input	Input	Input	Input	Output	Pre-Condition	Effect	Simila-rity	Matching Degree
	1	1	1	1	1	1	1		
WsRequest_Hotel	City	Spa	Three Star	Creditcard	Confirmation	Visa	Booking	0.869	Plugin
WsAdvertisement_Hotel_1	Area	Hotel Facility	No.Star	Creditcard	Confirmation	Payment	Available		
Path_Weight	1/2	1/3	1/3	1	1	1/2*1/4	1		
WsAdertisement _Hotel_2	Bang-kok	Health Club	-	-	Accept	Aeon	Confirm	0.000	Fail
Path_Weight	0	0	0	0	0	0	0		
WsAdvertisement_Hotel_3	Area	Sport Club	NO. Star	Visa	Booking	Payment	-	0.590	Subsume
Path_Weight	1/2	0	1/3	0	0	1/8	0		

6. Conclusion

In this paper, we propose a hybrid matching technique which supports semantic web service discovery to arrive at an extensible and efficient matchmaking by means of five fine-grained matching classification and adjustable threshold values. A straightforward case study was employed to demonstrate the viability of those proposed mechanisms. As such, relevant results were obtained which yielded higher level of user satisfaction.

7. References

[1] Fabien D., Zohra B., and Mathieu R, “A Context-Based Measure for Discovering Approximate Semantic Matching between Schema Elements”, Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM), 2006.

[2] Gongzhen W., Donghong X., Young Q. and Di H., "A Semantic Matching Algorithm for Web Services Based on Improved Semantic Distance", Proceedings of 4th International Conference on Next Generation Web Services Practices 2008, pp. 101-106.
[3] Li, L. and I. Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology", Proceedings of the Twelfth International (WWW 2003), Budapest, 2003, pp. 331-339.
[4] Matthias K., Benedikt F., and Katia S., "Automated Semantic Web Service Discovery with OWLS-MX", Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS 2006), May 8-12, 2006, pp. 915-922.
[5] Mehmet Senvar and Ayse Benser, "Matchmaking of Semantic Web Service Using Semantic-Distance Information", ADVIS 2006, pp. 177-186.
[6] Michael C., Daconta Leo J., and Kevin T. Smith, "The Semantic Web", Wiley Publishing, Inc., Canada, 2003.
[7] MinLiu., Wi Gao., Weiming Shen., and Qi Hao. "A Semantic-augmented multi-level matching model of Web services", CIMS Research Center, Tongji University, Shanghai, 2009, pp.205-215.
[8] Paolucci, M., T. Kawamura, T. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities", Proceedings of the First International Semantic Web Conference (ISWC), Sardinia, Italy, 2004, pp. 333-347.
[9] Sriharee, N., Senivongse, T., Teppaboot, C., and Futatsugi, K., "Adding Semantics to Attribute-Based Discovery of Web Services", Proceedings of International Symposium on Web Services and Applications (ISWS'04), Las Vegas, Nevada, USA., June 2004, pp. 790-794.
[10] Srinivasan, N., M. Paolucci and K. Sycara, "An Effidient Algorithm for OWL-S Based Semantic Search in UDDI", First Internatinal Workshop on Semantic Web Services and Web Process Composition (SWSWPC), vol. 3387 of LNCS 2004, pp. 96-110.
[11] Wang, Zhenqiang; Liu, Kaiyin; Lv, Guoying; Hao, and Xiaoyan, "Study of An Algorithm of Web Service Matching Based On Semantic Web Service", Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), pp. 429-433.
[12] OWL-S Submission, http://www.w3.org/Submission /OWL-S, 2004.